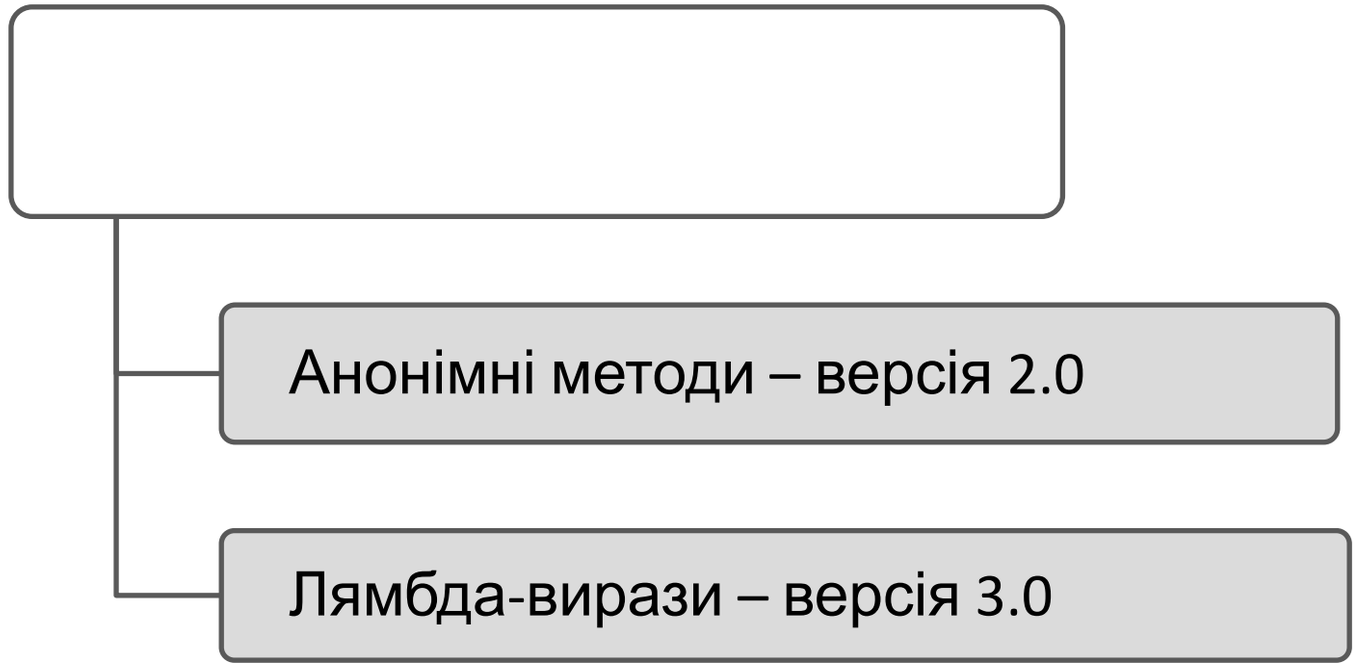
Лямбда-вирази

Анонімні методи -> лямбда-вирази
Одиночні та блокові лямбда-вирази
Узагальнені делегати та лямбда-вирази
Дерева виразів

Анонімні функції



Анонімні методи

Анонімний метод - один із способів створення безіменного блоку коду, пов'язаного з конкретним екземпляром делегата. Для створення анонімного методу досить вказати блок коду після ключового слова ***delegate***.

Приклад

```
delegate int Sum(int n);

static Sum SomeVar()
{
    int result = 0;

    //  Виклик анонімного метода
    Sum del = delegate (int n)
    {
        for (int i = 0; i <= n; i++)
            result += i;
        return result;
    };
    return del;
}
```

```
static void Main()
{
    Sum del1 = SomeVar();

    for (int i = 1; i <= 5; i++)
    {
        Console.WriteLine
        ("Сумма {0} равна: {1}",
         i, del1(i));
    }

    Console.ReadLine();
}
```

Результат

Сума 1 дорівнює: 1
Сума 2 дорівнює: 4
Сума 3 дорівнює:
10
Сума 4 дорівнює:
20
Сума 5 дорівнює:
35

Лямбда-вирази

Лямбда-вирази є спрощеним записом анонімних методів. Вони дозволяють створити повноцінні лаконічні методи, які можуть повертати деяке значення і які можна передати в якості параметрів в інші методи.

Лямбда-вирази

І анонімні методи, і лямбда-вирази дозволяють визначити вбудовану реалізацію методу, однак анонімний метод вимагає явно визначати типи параметрів і тип змінної, що повертається. Лямбда-вираз дозволяє компілятору логічно виводити тип змінної на основі контексту.

Лямбда-вирази

Синтаксис:

(список параметрів) => блок коду

Ліва частина може складатися з нуля або більше параметрів, далі записується символ лямбда => який застосовується для відділення оголошення параметрів від реалізації методу, що знаходиться в правій частині.

Приклад

```
List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7 };
    var evens = numbers.FindAll(n => n % 2 == 0);
    var evens2 = numbers.FindAll((int n) => { return n % 2 == 0; });
    foreach (int i in evens)
        Console.Write(i + " ");
    Console.WriteLine();
    foreach (int j in evens2)
        Console.Write(j + " ");
```

Лямбда-вирази

Якщо тіло лямбда-виразу складається з одного виразу, то утворюється *одиночний* лямбда-вираз. У цьому випадку тіло виразу не ставиться в фігурні дужки. Якщо ж тіло лямбда-виразу складається з блоку операторів, вкладених у фігурні дужки, то утворюється *блоковий* лямбда-вираз, який може містити цілий ряд операторів, в тому числі цикли, виклики методів і умовні оператори.

Одиночні лямбда-вирази

```
delegate bool InRange(int lower, int upper, int v);
// оголошується тип делегата, сумісного з цим лямбда-виразом

InRange rangeOK = (low, high, val) => val >= low && val <= high;
// створюється примірник делегата InRange


if(rangeOK(1, 5, 3))
    Console.WriteLine
    ("Число 3 знаходиться в межах від 1 до 5.");
// використання лямбда-виразу
```

Одиночні лямбда-вирази

```
// Створимо декілька делегатів,
// що імітують форму реєстації
delegate int LengthLogin(string s);
delegate bool BoolPassword(string s1, string s2);

private static void SetLogin()
{
    Console.Write("Введіть логін: ");
    string login = Console.ReadLine();
    // Використовуємо лямбда-вираз
    LengthLogin lengthLoginDelegate = s => s.Length;
    int lengthLogin = lengthLoginDelegate(login);
    if (lengthLogin > 25)
    {
        Console.WriteLine
        ("Занадто довге ім'я\n");
        // Рекурсія, щоб ввести логін заново
        SetLogin();
    }
}
```

```
static void Main()
{
    SetLogin();
    Console.Write("Введіть пароль: ");
    string password1 = Console.ReadLine();
        Console.Write("Повторіть пароль: ");
    string password2 = Console.ReadLine();
    // Використовуємо лямбда-вираз
    BoolPassword bp = (s1, s2) => s1 == s2;
    if (bp(password1, password2))
        Console.WriteLine
        ("Реєстрація завершена!");
    else
        Console.WriteLine
        ("Паролі не співпадають");

    Console.ReadLine();
}
```

Блокові лямбда-вирази

```
delegate int IntOp(int end);

IntOp lambda = n =>
            {
                int r = 1;
                for (int i = 1; i <= n; i++)
                    r = i * r;
                return r;
            };

Console.WriteLine("Факторіал 3 дорівнює " + lambda(3));
Console.WriteLine("Факторіал 5 дорівнює " + lambda(5));
```

Блокові лямбда-вирази

```
delegate void Captcha(string s1, string s2);
delegate bool BoolPassword(string s1, string s2);
static void Main()
{
    SetLogin();
    Console.Write("Перевірте пароль: ");
    string password1 = Console.ReadLine();
    Console.Write("Повторіть пароль: ");
    string password2 = Console.ReadLine();
    BoolPassword bp = (s1, s2) => s1 == s2;
    if (bp(password1, password2))
    {
        Random ran = new Random();
        string resCaptcha = "";
        for (int i = 0; i < 10; i++)
            resCaptcha += (char)ran.Next(0, 100);
        Console.WriteLine
        ("Введіть код xD: " + resCaptcha);
        string resCode = Console.ReadLine();
        Captcha cp = (s1, s2) =>
        {
            if (s1 == s2)
                Console.WriteLine
                ("Реєстрація
завершена!");
            else
                Console.WriteLine
                ("Неправильний код xD");
            return;
        };
        cp(resCaptcha, resCode);
    }
    else
        Console.WriteLine
        ("Паролі не співпадають");
    Console.ReadLine();
}
```

Лямбда-вирази

```
delegate void message();

static void Main(string[] args)
{
        message GetMessage = () =>
     { Console.WriteLine("Лямбда-вираз"); };

        GetMessage();

        Console.Read();
    }
```

Лямбда-вирази

```
delegate void message();

static void Main(string[] args)
     {
          message GetMessage = () => Show_Message();

          GetMessage();
     }
     private static void Show_Message()
     {
          Console.WriteLine("Привіт!");
     }
```

Узагальнені делегати та лямбда-вирази

```
List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7 };
Func<int, bool> where = n => n < 6;
Func<int, int> select = n => n;
Func<int, string> orderby = n => n % 2 == 0 ? "even" : "odd";
var nums = numbers.Where(where).OrderBy(orderby).Select(select);
foreach (int i in nums)
    Console.Write(i + ", ");
```

Результат: 2, 4, 1, 3,
5

Дерева виразів

Дерево виразів є лямбда-виразом у вигляді даних. Це означає, що сам лямбда-вираз не можна виконати, але можна перетворити в виконувану форму.

System.Linq.Expressions.Expression<TDelegate>

Приклад

```
// Представляємо лямбда-вираз у вигляді даних.
    Expression<Func<int, int, bool>> IsFactorExp =
(n, d) => (d != 0)? (n % d) == 0: false;


    // Компілюємо дані виразу в виконуваний код.
    Func<int, int, bool> IsFactor = IsFactorExp.Compile();


    // Застосовуємо вираз
    if (IsFactor(10, 5))
        Console.WriteLine("Число 5 є множником 10.");
    if (!IsFactor(10, 7))
        Console.WriteLine("Число 7 не є множником 10.");
    Console.WriteLine();
```